

## Traffic Flow Measurement: Meter MIB

### Status of this Memo

This memo defines an Experimental Protocol for the Internet community. This memo does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

### Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, this memo defines managed objects used for obtaining traffic flow information from network traffic meters.

### Table of Contents

1	The Network Management Framework . . . . .	1
2	Objects . . . . .	2
2.1	Format of Definitions . . . . .	3
3	Overview . . . . .	3
3.1	Scope of Definitions, Textual Conventions . . . . .	3
3.2	Usage of the MIB variables . . . . .	4
4	Definitions . . . . .	6
5	Acknowledgements . . . . .	37
6	References . . . . .	37
7	Security Considerations . . . . .	38
8	Author's Address . . . . .	38

### 1 The Network Management Framework

The Internet-standard Network Management Framework consists of three components. They are:

RFC 1155 defines the SMI, the mechanisms used for describing and naming objects for the purpose of management. STD 16, RFC 1212 defines a more concise description mechanism, which is wholly consistent with the SMI.

RFC 1156 defines MIB-I, the core set of managed objects for the Internet suite of protocols. STD 17, RFC 1213 [1] defines MIB-II, an evolution of MIB-I based on implementation experience and new operational requirements.

STD 15, RFC 1157 defines the SNMP, the protocol used for network access to managed objects.

RFC 1442 [2] defines the SMI for version 2 of the Simple Network Management Protocol.

RFCs 1443 and 1444 [3,4] define Textual Conventions and Conformance Statements for version 2 of the Simple Network Management Protocol.

RFC 1452 [5] describes how versions 1 and 2 of the Simple Network Management Protocol should coexist.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

## 2 Objects

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [6] defined in the SMI. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to the object type.

The syntax of an object type defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the SMI [2] purposely restricts the ASN.1 constructs which may be used. These restrictions are explicitly made for simplicity.

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network.

The SMI specifies the use of the basic encoding rules of ASN.1 [7], subject to the additional requirements imposed by the SNMP.

## 2.1 Format of Definitions

Section 4 contains the specification of all object types contained in this MIB module. These object types are defined using the conventions defined in [2] and [3].

## 3 Overview

Traffic Flow Measurement seeks to provide a well-defined method for gathering traffic flow information from networks and internetworks. The background for this is given in "Traffic Flow Measurement: Background" [8]. The Realtime Traffic Flow Measurement (rtfm) Working Group has produced a measurement architecture to achieve it; this is documented in "Traffic Flow Measurement: Architecture" [9]. The architecture defines three entities:

- METERS, which observe network traffic flows and build up a table of flow data records for them,
- METER READERS, which collect traffic flow data from meters, and
- MANAGERS, which oversee the operation of meters and meter readers.

This memo defines the SNMP management information for a Traffic Flow Meter (TFM). It documents the earlier work of the Internet Accounting Working Group, and is intended to provide a starting point for the Realtime Traffic Flow Measurement Working Group.

### 3.1 Scope of Definitions, Textual Conventions

All objects defined in this memo are registered in a single subtree within the mib-2 namespace [1,2], and are for use in network devices which may perform a PDU forwarding or monitoring function. For these devices, the value of the ifSpecific variable in the MIB-II [1] has the OBJECT IDENTIFIER value:

flowMIB OBJECT IDENTIFIER ::= mib-2 40

as defined below.

The RTFM Meter MIB was first produced and tested using SNMPv1. It has been converted into SNMPv2 following the guidelines in RFC 1452 [5].

### 3.2 Usage of the MIB variables

The MIB breaks into four parts - control, flows, rules and conformance statements.

The rules implement the minimum set of packet-matching actions, as set out in the "Traffic Flow Measurement: Architecture" document [9]. In addition they provide for BASIC-style subroutines, allowing a network manager to dramatically reduce the number of rules required to monitor a big network.

Traffic flows are identified by a set of attributes for each of its end-points. Attributes include network addresses for each layer of the network protocol stack, and 'subscriber ids,' which may be used to identify an accountable entity for the flow.

The conformance statements are set out as defined in [4]. They explain what must be implemented in a meter which claims to conform to this MIB.

To retrieve flow data one could simply do a linear scan of the flow table. This would certainly work, but would require a lot of protocol exchanges. To reduce the overhead in retrieving flow data the flow table uses a TimeFilter variable, defined as a Textual Convention in the RMON2 MIB [10]. This, when used together with SNMPv2's GetBulk request, allows a meter reader to scan the flow table and upload a specified set of flow attributes for those rows which have changed since the last reading.

As an alternative method of reading flow data, the MIB provides an index into the flow table called flowColumnActivityTable. This is (logically) a three-dimensional array, subscripted by flow attribute, activity time and starting flow number. This allows a meter reader to retrieve (in an opaque object) data for a column of the flow table with a minimum of SNMP overhead. An attempt has been made to include a full ASN.1 definition of the flowColumnActivityData object.

One aspect of data collection which needs emphasis is that all the MIB variables are set up to allow multiple independent collectors to work properly, i.e. the flow table indexes are stateless. An alternative approach would have been to 'snapshot' the flow table, which would mean that the meter readers would have to be synchronized. The stateless approach does mean that two meter readers will never return exactly the same set of traffic counts, but over long periods (e.g. 15-minute collections over a day) the discrepancies are acceptable. If one really needs a snapshot, this can be achieved by switching to an identical rule set with a different RuleSet number, hence asynchronous collections may be

regarded as a useful generalisation of synchronised ones.

The control variables are the minimum set required for a meter reader. Their number has been whittled down as experience has been gained with the MIB implementation. A few of them are 'general,' i.e. they control the overall behaviour of the meter. These are set by a single 'master' manager, and no other manager should attempt to change their values. The decision as to which manager is the 'master' must be made by the network operations personnel responsible; this MIB does not attempt to provide any support for interaction between managers.

There are three other groups of control groups, arranged into tables in the same way as in the RMON MIB [10]. They are used as follows:

- RULE SET INFO: Before attempting to download a rule table a manager must create a row in the flowRuleSetInfo with flowRuleInfoStatus set to 'createAndWait.' When the rule set is ready the manager must set RuleSetInfo to 'active,' indicating that the rule set is ready for use.
- METER READER INFO: Any meter reader wishing to collect data reliably for all flows should first create a row in the flowReaderInfoTable with flowReaderStatus set to 'active.' It should write that row's flowReaderLastTime object each time it starts a collection pass through the flow table. The meter will not recover a flow's memory until every meter reader holding a row in this table has collected that flow's data.
- MANAGER INFO: Any manager wishing to download rule sets to the meter must create a row in the flowManagerInfo table with flowManagerStatus set to 'active.'. Once it has a table row, the manager may set the control variables in its row so as to cause the meter to run any valid rule set held by the meter.

## 4 Definitions

```
FLOW-METER-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, Counter32, Integer32, TimeTicks
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowStatus, TimeStamp
        FROM SNMPv2-TC
    OBJECT-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF
    mib-2, ifIndex
        FROM RFC1213-MIB
    OwnerString
        FROM RMON-MIB;
```

```
flowMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "9603080208Z"
    ORGANIZATION "IETF Realtime Traffic Flow Measurement Working Group"
    CONTACT-INFO
        "Nevil Brownlee, The University of Auckland
         Email: n.brownlee@auckland.ac.nz"
    DESCRIPTION
        "MIB for the RTFM Traffic Flow Meter."
    ::= { mib-2 40 }
```

```
flowControl          OBJECT IDENTIFIER ::= { flowMIB 1 }
```

```
flowData             OBJECT IDENTIFIER ::= { flowMIB 2 }
```

```
flowRules            OBJECT IDENTIFIER ::= { flowMIB 3 }
```

```
flowMIBConformance  OBJECT IDENTIFIER ::= { flowMIB 4 }
```

```
-- Textual Conventions
```

```
TimeFilter ::= TEXTUAL-CONVENTION
```

```
    STATUS current
```

```
    DESCRIPTION
```

```
        "Used as an index to a table.  A TimeFilter variable allows
         a GetNext or GetBulk request to find rows in a table for
         which the TimeFilter index variable is greater than or equal
         to a specified value.  For example, a meter reader could
         find all rows in the flow table which have been active at or
         since a specified time."
```

More details on TimeFilter variables, their implementation and use can be found in the RMON2 MIB [10]."

SYNTAX TimeTicks

AddressType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Indicates the type of an adjacent address or peer address.

The values used are from the 'Address Family Numbers' section of the Assigned Numbers RFC [11]."

SYNTAX INTEGER {

ip(1),

nsap(3),

ieee802(6),

ipx(11),

appletalk(12),

decnet(13) }

AdjacentAddress ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Specifies the value of an adjacent address for various media. The values used for IEEE 802 media are from the 'Network Management Parameters (ifType definitions)' section of the Assigned Numbers RFC [11]. Address format depends on the actual media, as follows:

Ethernet: ethernet(7)

6-octet 802.3 MAC address in 'canonical' order

FDDI: fddi(15)

FddiMACLongAddress, i.e. a 6-octet MAC address

in 'canonical' order (defined in the FDDI MIB [12])

Token Ring: tokenring(9)

6-octet 802.5 MAC address in 'canonical' order

PeerAddress: other(1)

If traffic is being metered inside a tunnel, its adjacent addresses will be the peer addresses of hosts at the ends of the tunnel

"

SYNTAX OCTET STRING (SIZE (6..20))

PeerAddress ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Specifies the value of a peer address for various network

protocols. Address format depends on the actual protocol,  
as follows:

IP:                ip(1)  
      4-octet IpAddress (defined in the SNMPv2 SMI [2])

CLNS:             nsap(3)  
      NsapAddress (defined in the SNMPv2 SMI [2])

Novell:           ipx(11)  
      4-octet Network number,  
      6-octet Host number (MAC address)

AppleTalk:       appletalk(12)  
      2-octet Network number (sixteen bits),  
      1-octet Host number (eight bits)

DECnet:           decnet(13)  
      1-octet Area number (in low-order six bits),  
      2-octet Host number (in low-order ten bits)

"

SYNTAX OCTET STRING (SIZE (3..20))

TransportAddress ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Specifies the value of a transport address for various  
network protocols. Format as follows:

IP:  
      2-octet UDP or TCP port number

Other protocols:  
      2-octet port number

"

SYNTAX OCTET STRING (SIZE (2))

RuleAddress ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Specifies the value of an address. Is a superset of  
AdjacentAddress, PeerAddress and TransportAddress."

SYNTAX OCTET STRING (SIZE (2..20))

FlowAttributeNumber ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Uniquely identifies an attribute within a flow data record."



```
SYNTAX  INTEGER {
    flowIndex(1),
    flowStatus(2),
    flowTimeMark(3),

    sourceInterface(4),
    sourceAdjacentType(5),
    sourceAdjacentAddress(6),
    sourceAdjacentMask(7),
    sourcePeerType(8),
    sourcePeerAddress(9),
    sourcePeerMask(10),
    sourceTransType(11),
    sourceTransAddress(12),
    sourceTransMask(13),

    destInterface(14),
    destAdjacentType(15),
    destAdjacentAddress(16),
    destAdjacentMask(17),
    destPeerType(18),
    destPeerAddress(19),
    destPeerMask(20),
    destTransType(21),
    destTransAddress(22),
    destTransMask(23),

    pduScale(234),
    octetScale(25),

    ruleSet(26),
    toOctets(27),           -- Source-to-Dest
    toPDUs(28),
    fromOctets(29),         -- Dest-to-Source
    fromPDUs(30),
    firstTime(31),         -- Activity times
    lastActiveTime(32),

    sourceSubscriberID(33), -- Subscriber ID
    destSubscriberID(34),
    sessionID(35),

    sourceClass(36),        -- Computed attributes
    destClass(37),
    flowClass(38),
    sourceKind(39),
    destKind(40),
    flowKind(41) }
```

RuleAttributeNumber ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Uniquely identifies an attribute which may be tested in a rule. These include attributes whose values come directly from the flow's packets and the five 'meter' variables used to hold an AttributeValue. Attributes derived from the rules - e.g. address masks - may not be tested."

SYNTAX INTEGER {

```

null(0),
sourceInterface(4),          -- Source Address
sourceAdjacentType(5),
sourceAdjacentAddress(6),
sourcePeerType(8),
sourcePeerAddress(9),
sourceTransType(11),
sourceTransAddress(12),
destInterface(14),          -- Dest Address
destAdjacentType(15),
destAdjacentAddress(16),
destPeerType(18),
destPeerAddress(19),
destTransType(21),
destTransAddress(22),
sourceSubscriberID(33),     -- Subscriber ID
destSubscriberID(34),
sessionID(35),
v1(51),                    -- Meter variables
v2(52),
v3(53),
v4(54),
v5(55) }
```

ActionNumber ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Uniquely identifies the action of a rule, i.e. the Pattern Matching Engine's opcode number. Details of the opcodes are given in the 'Traffic Flow Measurement: Architecture' document [9]."

SYNTAX INTEGER {

```

ignore(1),
fail(2),
count(3),
countPkt(4),
return(5),
gosub(6),
gosubAct(7),
```

```

    assign(8),
    assignAct(9),
    goto(10),
    gotoAct(11),
    pushRuleTo(12),
    pushRuleToAct(13),
    pushPktTo(14),
    pushPktToAct(15) }

```

```

--
-- Control Group: Rule Set Info Table
--

```

```

flowRuleSetInfoTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FlowRuleSetInfoEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An array of information about the rule sets held in the
        meter. Rule set 1 is the meter default, used when the meter
        starts up. It is built in to the meter; it may not be
        changed."
    ::= { flowControl 1 }

```

```

flowRuleSetInfoEntry OBJECT-TYPE
    SYNTAX FlowRuleSetInfoEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Information about a particular rule set."
    INDEX { flowRuleInfoIndex }
    ::= { flowRuleSetInfoTable 1 }

```

```

FlowRuleSetInfoEntry ::= SEQUENCE {
    flowRuleInfoIndex      Integer32,
    flowRuleInfoSize       Integer32,
    flowRuleInfoOwner      OwnerString,
    flowRuleInfoTimeStamp  TimeStamp,
    flowRuleInfoStatus     RowStatus
}

```

```

flowRuleInfoIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An index which selects an entry in the flowRuleSetInfoTable.

```

Each such entry contains control information for a particular rule set which the meter may run."

::= { flowRuleSetInfoEntry 1 }

flowRuleInfoSize OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Number of rules in this rule set. Setting this variable will cause the meter to allocate space for these rules."

::= { flowRuleSetInfoEntry 2 }

flowRuleInfoOwner OBJECT-TYPE

SYNTAX OwnerString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Identifies the manager which configured this rule set."

::= { flowRuleSetInfoEntry 3 }

flowRuleInfoTimeStamp OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Time this rule set was last changed."

::= { flowRuleSetInfoEntry 4 }

flowRuleInfoStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this rule set. If this object's value is not active(1), the meter must not attempt to use this rule set."

::= { flowRuleSetInfoEntry 5 }

--

-- Control Group: Interface Info Table

--

flowInterfaceTable OBJECT-TYPE

SYNTAX SEQUENCE OF FlowInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An array of information specific to each meter interface."  
 ::= { flowControl 2 }

flowInterfaceEntry OBJECT-TYPE

SYNTAX FlowInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Information about a particular interface."

INDEX { ifIndex }

::= { flowInterfaceTable 1 }

FlowInterfaceEntry ::= SEQUENCE {

flowInterfaceRate Integer32,

flowInterfaceLostPackets Counter32

}

flowInterfaceRate OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The parameter N for statistical counting on this interface.

Set to N to count 1/Nth of the packets appearing at this

interface. A meter should choose its own algorithm to

introduce variance into the sampling so that exactly every Nth

packet is not counted. A sampling rate of 1 counts all

packets. A sampling rate of 0 results in the interface

being ignored by the meter."

::= { flowInterfaceEntry 1 }

flowInterfaceLostPackets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets the meter has lost for this interface.

Such losses may occur because the meter has been unable to

keep up with the traffic volume."

::= { flowInterfaceEntry 2 }

--

-- Control Group: Meter Reader Info Table

--

-- Any meter reader wishing to collect data reliably for all flows

-- should first create a row in this table. It should write that

-- row's flowReaderLastTime object each time it starts a collection

```
-- pass through the flow table.

-- The meter will not recover a flow's memory until every meter reader
-- holding a row in this table has collected that flow's data.

-- If a meter reader does not create a row in this table, e.g. because
-- it failed authentication in the meter's SNMP write community,
-- collection can still proceed but the meter may not be able to
-- recover inactive flows.
```

flowReaderInfoTable OBJECT-TYPE

SYNTAX SEQUENCE OF FlowReaderInfoEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An array of information about meter readers which have  
registered their intent to collect flow data from this meter."

::= { flowControl 3 }

flowReaderInfoEntry OBJECT-TYPE

SYNTAX FlowReaderInfoEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Information about a particular meter reader."

INDEX { flowReaderIndex }

::= { flowReaderInfoTable 1 }

FlowReaderInfoEntry ::= SEQUENCE {

flowReaderIndex Integer32,

flowReaderTimeout Integer32,

flowReaderOwner OwnerString,

flowReaderLastTime TimeStamp,

flowReaderPreviousTime TimeStamp,

flowReaderStatus RowStatus

}

flowReaderIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Selects an entry from the array of meter reader info entries."

::= { flowReaderInfoEntry 1 }

flowReaderTimeout OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Specifies the maximum time (in seconds) between flow data collections for this meter reader. If this time elapses without a collection, the meter should assume that this meter reader has stopped collecting, and delete this row from the table."

::= { flowReaderInfoEntry 2 }

flowReaderOwner OBJECT-TYPE

SYNTAX OwnerString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Identifies the meter reader which created this row."

::= { flowReaderInfoEntry 3 }

flowReaderLastTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Time this meter reader began its most recent data collection."

This variable should be written by a meter reader as the first step in reading flow data. The meter will set this LastTime value to sysUptime and set its PreviousTime value (below) to the old LastTime. This allows the meter to recover flows which have been inactive since PreviousTime, for these have been collected at least once.

If the meter fails to write flowLastReadTime, e.g. by failing authentication in the meter's SNMP write community, collection may still proceed but the meter may not be able to recover inactive flows."

::= { flowReaderInfoEntry 4 }

flowReaderPreviousTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Time this meter reader began the collection before last."

::= { flowReaderInfoEntry 5 }

flowReaderStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

```

STATUS    current
DESCRIPTION
    "The status of this meter reader."
 ::= { flowReaderInfoEntry 6 }

```

```

--
-- Control Group:  Manager Info Table
--

```

```

-- Any manager wishing to download rule sets to the meter must create
-- a row in this table.  Once it has a table row, the manager may set
-- the control variables in its row so as to cause the meter to run
-- any valid rule set held by the meter.

```

```

flowManagerInfoTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FlowManagerInfoEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
        "An array of information about managers which have
         registered their intent to run rule sets on this meter."
    ::= { flowControl 4 }

```

```

flowManagerInfoEntry OBJECT-TYPE
    SYNTAX  FlowManagerInfoEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
        "Information about a particular meter reader."
    INDEX   { flowManagerIndex }
    ::= { flowManagerInfoTable 1 }

```

```

FlowManagerInfoEntry ::= SEQUENCE {
    flowManagerIndex          Integer32,
    flowManagerCurrentRuleSet Integer32,
    flowManagerStandbyRuleSet Integer32,
    flowManagerHighWaterMark  INTEGER,
    flowManagerCounterWrap    INTEGER,
    flowManagerOwner           OwnerString,
    flowManagerTimeStamp       TimeStamp,
    flowManagerStatus          RowStatus
}

```

```

flowManagerIndex OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION

```



"Selects an entry from the array of manager info entries."  
 ::= { flowManagerInfoEntry 1 }

flowManagerCurrentRuleSet OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Index to the array of rule sets. Specifies which set of rules is currently being used for accounting by this manager. When the manager sets this variable the meter will close its current rule set and start using the new one. Flows created by the old rule set remain in memory, orphaned until their data has been read. Specifying rule set 0 (the empty set) stops flow measurement by this manager."

::= { flowManagerInfoEntry 2 }

flowManagerStandbyRuleSet OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Index to the array of rule sets. After reaching HighWaterMark (see below) the manager may switch to using its standby rule set. For this to be effective the manager should have downloaded a standby rule set which uses a coarser reporting granularity. The manager may also need to decrease the meter reading interval so that the meter can recover flows measured by its normal rule set."

DEFVAL { 0 } -- No standby

::= { flowManagerInfoEntry 3 }

flowManagerHighWaterMark OBJECT-TYPE

SYNTAX INTEGER (0..100)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A value expressed as a percentage, interpreted by the meter as an indication of how full the flow table should be before it should switch to the standby rule set (if one has been specified) for this manager. Values of 0% or 100% disable the checking represented by this variable."

::= { flowManagerInfoEntry 4 }

flowManagerCounterWrap OBJECT-TYPE

SYNTAX INTEGER { wrap(1), scale(2) }

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"Specifies whether PDU and octet counters should wrap when they reach the top of their range (normal behaviour for Counter32 objects), or whether their scale factors should be used instead. The combination of counter and scale factor allows counts to be returned as binary floating point numbers, with 32-bit mantissas and 8-bit exponents."

DEFVAL { wrap }

::= { flowManagerInfoEntry 5 }

## flowManagerOwner OBJECT-TYPE

SYNTAX OwnerString

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"Identifies the manager which created this row."

::= { flowManagerInfoEntry 6 }

## flowManagerTimeStamp OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"Time this row was last changed by its manager."

::= { flowManagerInfoEntry 7 }

## flowManagerStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"The status of this manager."

::= { flowManagerInfoEntry 8 }

--

-- Control Group: General Meter Control Variables

--

-- At present the meter only runs a single rule set - the 'current'  
 -- one and has a single 'standby' rule set. In future it may be  
 -- developed so as to run multiple rule sets simultaneously; that would  
 -- require a more elaborate set of control variables to allow reliable  
 -- operation.

## flowFloodMark OBJECT-TYPE

SYNTAX INTEGER (0..100)

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"A value expressed as a percentage, interpreted by the meter as an indication of how full the flow table should be before it should take some action to avoid running out of resources to handle new flows. Values of 0% or 100% disable the checking represented by this variable."

::= { flowControl 5 }

## flowInactivityTimeout OBJECT-TYPE

SYNTAX Integer32 (1..3600)

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The time in seconds since the last packet seen, after which the flow may be terminated. Note that although a flow may have been terminated, its data must be collected before its memory can be recovered."

DEFVAL { 600 } -- 10 minutes

::= { flowControl 6 }

## flowActiveFlows OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The numbers of flows which are currently in use, i.e. have been active since the last collection."

::= { flowControl 7 }

## flowMaxFlows OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The maximum number of flows allowed in the meter's flow table. At present this is determined when the meter is first started up."

::= { flowControl 8 }

--

-- The Flow Table

--

-- This is a table kept by a meter, with one flow data entry for every  
 -- flow being measured. Each flow data entry stores the attribute  
 -- values for a traffic flow. Details of flows and their attributes  
 -- are given in the 'Traffic Flow Measurement: Architecture'

```
-- document [9].

-- From time to time a meter reader may sweep the flow table so as
-- to read counts. This is most effectively achieved by using the
-- TimeMark variable together with successive GetBulk requests to
-- retrieve the values of the desired flow attribute variables.

-- This scheme allows multiple meter readers to independently use the
-- same meter; the meter readers do not have to be synchronised and
-- they may use different collection intervals.
```

```
flowDataTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FlowDataEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The list of all flows being measured."
    ::= { flowData 1 }
```

```
flowDataEntry OBJECT-TYPE
    SYNTAX FlowDataEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The flow data record for a particular flow."
    INDEX { flowDataTimeMark, flowDataIndex }
    ::= { flowDataTable 1 }
```

```
FlowDataEntry ::= SEQUENCE {
    flowDataIndex                Integer32,
    flowDataTimeMark             TimeFilter,
    flowDataStatus               INTEGER,

    flowDataSourceInterface      Integer32,
    flowDataSourceAdjacentType   AddressType,
    flowDataSourceAdjacentAddress AdjacentAddress,
    flowDataSourceAdjacentMask   AdjacentAddress,
    flowDataSourcePeerType       AddressType,
    flowDataSourcePeerAddress    PeerAddress,
    flowDataSourcePeerMask       PeerAddress,
    flowDataSourceTransType      INTEGER,
    flowDataSourceTransAddress   TransportAddress,
    flowDataSourceTransMask      TransportAddress,

    flowDataDestInterface        Integer32,
    flowDataDestAdjacentType     AddressType,
    flowDataDestAdjacentAddress  AdjacentAddress,
    flowDataDestAdjacentMask     AdjacentAddress,
```

```

flowDataDestPeerType      AddressType,
flowDataDestPeerAddress   PeerAddress,
flowDataDestPeerMask      PeerAddress,
flowDataDestTransType     INTEGER,
flowDataDestTransAddress  TransportAddress,
flowDataDestTransMask     TransportAddress,

flowDataPDUScale          INTEGER,
flowDataOctetScale        INTEGER,

flowDataRuleSet           INTEGER,

flowDataToOctets          Counter32,      -- Source->Dest
flowDataToPDUs            Counter32,
flowDataFromOctets        Counter32,      -- Dest->Source
flowDataFromPDUs          Counter32,
flowDataFirstTime         TimeTicks,      -- Activity times
flowDataLastActiveTime    TimeTicks,

flowDataSourceSubscriberID OCTET STRING,
flowDataDestSubscriberID  OCTET STRING,
flowDataSessionID         OCTET STRING,

flowDataSourceClass       INTEGER,
flowDataDestClass         INTEGER,
flowDataClass             INTEGER,
flowDataSourceKind        INTEGER,
flowDataDestKind          INTEGER,
flowDataKind              INTEGER
}

```

#### flowDataIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Value of this flow data record's index within the meter's flow table."

::= { flowDataEntry 1 }

#### flowDataTimeMark OBJECT-TYPE

SYNTAX TimeFilter

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A TimeFilter for this entry. Allows GetNext and GetBulk to find flow table rows which have changed since a specified value of sysUptime."

```
::= { flowDataEntry 2 }
```

flowDataStatus OBJECT-TYPE

```
SYNTAX  INTEGER { inactive(1), current(2), idle(3) }
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "Status of this flow data record."
```

```
::= { flowDataEntry 3 }
```

flowDataSourceInterface OBJECT-TYPE

```
SYNTAX  Integer32
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "Index of the interface associated with the source address  
    for this flow.  It's value is one of those contained in the  
    ifIndex field of the meter's interfaces table."
```

```
::= { flowDataEntry 4 }
```

flowDataSourceAdjacentType OBJECT-TYPE

```
SYNTAX  AddressType
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "Adjacent address type of the source for this flow.  If  
    accounting is being performed at the network level the  
    adjacent address will probably be an 802 MAC address, and  
    the adjacent address type will indicate the medium type."
```

```
::= { flowDataEntry 5 }
```

flowDataSourceAdjacentAddress OBJECT-TYPE

```
SYNTAX  AdjacentAddress
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "Address of the adjacent device on the path for the source  
    for this flow."
```

```
::= { flowDataEntry 6 }
```

flowDataSourceAdjacentMask OBJECT-TYPE

```
SYNTAX  AdjacentAddress
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "1-bits in this mask indicate which bits must match when  
    comparing the adjacent source address for this flow."
```

```
::= { flowDataEntry 7 }
```

```
flowDataSourcePeerType OBJECT-TYPE
    SYNTAX  AddressType
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Peer address type of the source for this flow."
    ::= { flowDataEntry 8 }

flowDataSourcePeerAddress OBJECT-TYPE
    SYNTAX  PeerAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Address of the peer device for the source of this flow."
    ::= { flowDataEntry 9 }

flowDataSourcePeerMask OBJECT-TYPE
    SYNTAX  PeerAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
        comparing the source peer address for this flow."
    ::= { flowDataEntry 10 }

flowDataSourceTransType OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Transport address type of the source for this flow.  The
        value of this attribute will depend on the peer address type."
    ::= { flowDataEntry 11 }

flowDataSourceTransAddress OBJECT-TYPE
    SYNTAX  TransportAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Transport address for the source of this flow."
    ::= { flowDataEntry 12 }

flowDataSourceTransMask OBJECT-TYPE
    SYNTAX  TransportAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
```

comparing the transport source address for this flow."  
 ::= { flowDataEntry 13 }

flowDataDestInterface OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Index of the interface associated with the dest address for this flow. This value is one of the values contained in the ifIndex field of the interfaces table."

::= { flowDataEntry 14 }

flowDataDestAdjacentType OBJECT-TYPE

SYNTAX AddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Adjacent address type of the destination for this flow."

::= { flowDataEntry 15 }

flowDataDestAdjacentAddress OBJECT-TYPE

SYNTAX AdjacentAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Address of the adjacent device on the path for the destination for this flow."

::= { flowDataEntry 16 }

flowDataDestAdjacentMask OBJECT-TYPE

SYNTAX AdjacentAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"1-bits in this mask indicate which bits must match when comparing the adjacent dest address for this flow."

::= { flowDataEntry 17 }

flowDataDestPeerType OBJECT-TYPE

SYNTAX AddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Peer address type of the destination for this flow."

::= { flowDataEntry 18 }

flowDataDestPeerAddress OBJECT-TYPE



SYNTAX PeerAddress  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "Address of the peer device for the destination of this flow."  
 ::= { flowDataEntry 19 }

flowDataDestPeerMask OBJECT-TYPE  
SYNTAX PeerAddress  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "1-bits in this mask indicate which bits must match when  
    comparing the dest peer type for this flow."  
 ::= { flowDataEntry 20 }

flowDataDestTransType OBJECT-TYPE  
SYNTAX INTEGER (1..255)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "Transport address type of the destination for this flow. The  
    value of this attribute will depend on the peer address type."  
 ::= { flowDataEntry 21 }

flowDataDestTransAddress OBJECT-TYPE  
SYNTAX TransportAddress  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "Transport address for the destination of this flow."  
 ::= { flowDataEntry 22 }

flowDataDestTransMask OBJECT-TYPE  
SYNTAX TransportAddress  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "1-bits in this mask indicate which bits must match when  
    comparing the transport destination address for this flow."  
 ::= { flowDataEntry 23 }

flowDataPDUScale OBJECT-TYPE  
SYNTAX INTEGER (1..255)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The scale factor applied to this particular flow. Indicates

the number of bits the PDU counter values should be moved left to obtain the actual values."

::= { flowDataEntry 24 }

flowDataOctetScale OBJECT-TYPE

SYNTAX INTEGER (1..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The scale factor applied to this particular flow. Indicates the number of bits the octet counter values should be moved left to obtain the actual values."

::= { flowDataEntry 25 }

flowDataRuleSet OBJECT-TYPE

SYNTAX INTEGER (1..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The RuleSet number of the rule set which created this flow."

::= { flowDataEntry 26 }

flowDataToOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The count of octets flowing from source to dest address and being delivered to the protocol level being metered. In the case of IP this would count the number of octets delivered to the IP level."

::= { flowDataEntry 27 }

flowDataToPDUs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The count of protocol packets flowing from source to dest address and being delivered to the protocol level being metered. In the case of IP, for example, this would count the IP packets delivered to the IP protocol level."

::= { flowDataEntry 28 }

flowDataFromOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The count of octets flowing from dest to source address and being delivered to the protocol level being metered."

::= { flowDataEntry 29 }

## flowDataFromPDUs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The count of protocol packets flowing from dest to source address and being delivered to the protocol level being metered. In the case of IP, for example, this would count the IP packets delivered to the IP protocol level."

::= { flowDataEntry 30 }

## flowDataFirstTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The time at which this flow was first entered in the table"

::= { flowDataEntry 31 }

## flowDataLastActiveTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The last time this flow had activity, i.e. the time of arrival of the most recent PDU belonging to this flow."

::= { flowDataEntry 32 }

## flowDataSourceSubscriberID OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (4..20))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"Subscriber ID associated with the source address for this flow."

::= { flowDataEntry 33 }

## flowDataDestSubscriberID OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (4..20))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"Subscriber ID associated with the dest address for this

```
        flow."
 ::= { flowDataEntry 34 }

flowDataSessionID OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (4..10))
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Session ID for this flow.  Such an ID might be allocated
        by a network access server to distinguish a series of sessions
        between the same pair of addresses, which would otherwise
        appear to be parts of the same accounting flow."
 ::= { flowDataEntry 35 }

flowDataSourceClass OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Source class for this flow.  Determined by the rules, set by
        a PushRule action when this flow was entered in the table."
 ::= { flowDataEntry 36 }

flowDataDestClass OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Destination class for this flow.  Determined by the rules, set
        by a PushRule action when this flow was entered in the table."
 ::= { flowDataEntry 37 }

flowDataClass OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Class for this flow.  Determined by the rules, set by a
        PushRule action when this flow was entered in the table."
 ::= { flowDataEntry 38 }

flowDataSourceKind OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Source kind for this flow.  Determined by the rules, set by
        a PushRule action when this flow was entered in the table."
```

```
::= { flowDataEntry 39 }
```

```
flowDataDestKind OBJECT-TYPE
```

```
SYNTAX INTEGER (1..255)
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Destination kind for this flow. Determined by the rules, set
  by a PushRule action when this flow was entered in the table."
```

```
::= { flowDataEntry 40 }
```

```
flowDataKind OBJECT-TYPE
```

```
SYNTAX INTEGER (1..255)
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Class for this flow. Determined by the rules, set by a
  PushRule action when this flow was entered in the table."
```

```
::= { flowDataEntry 41 }
```

```
--
```

```
-- The Activity Column Table
```

```
--
```

```
flowColumnActivityTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF FlowColumnActivityEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Index into the Flow Table. Allows a meter reader to retrieve
  a list containing the flow table indices of flows which were
  last active at or after a given time, together with the values
  of a specified attribute for each such flow."
```

```
::= { flowData 2 }
```

```
flowColumnActivityEntry OBJECT-TYPE
```

```
SYNTAX FlowColumnActivityEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The Column Activity Entry for a particular attribute,
  activity time and flow."
```

```
INDEX { flowColumnActivityAttribute, flowColumnActivityTime,
        flowColumnActivityIndex }
```

```
::= { flowColumnActivityTable 1 }
```

```
FlowColumnActivityEntry ::= SEQUENCE {
```

```
flowColumnActivityAttribute  FlowAttributeNumber,
flowColumnActivityTime      TimeFilter,
flowColumnActivityIndex      Integer32,
flowColumnActivityData       OCTET STRING
}
```

flowColumnActivityAttribute OBJECT-TYPE

SYNTAX FlowAttributeNumber

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Specifies the attribute for which values are required from active flows."

::= { flowColumnActivityEntry 1 }

flowColumnActivityTime OBJECT-TYPE

SYNTAX TimeFilter

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable is a copy of flowDataLastActiveTime in the flow data record identified by the flowColumnActivityIndex value of this flowColumnActivityTable entry."

::= { flowColumnActivityEntry 2 }

flowColumnActivityIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Index of a flow table entry which was active at or after a specified flowColumnActivityTime."

::= { flowColumnActivityEntry 3 }

flowColumnActivityData OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (3..1000))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Collection of attribute data for flows active after flowColumnActivityTime. Within the OCTET STRING is a sequence of { flow index, attribute value } pairs, one for each active flow. The end of the sequence is marked by a flow index value of 0, indicating that there are no more rows in this column."

The format of objects inside flowColumnFlowData is as follows. All numbers are unsigned. Numbers and strings appear with

their high-order bytes leading. Numbers are fixed size, as specified by their SYNTAX in the flow table (above), i.e. one octet for flowAddressType and small constants, and four octets for Counter and Timeticks. Strings are variable-length, with the length given in a single leading octet.

The following is an attempt at an ASN.1 definition of flowColumnActivityData:

```
flowColumnActivityData ::= SEQUENCE flowRowItemEntry
flowRowItemEntry ::= SEQUENCE {
    flowRowNumber    INTEGER (1..65535),
                        -- 0 indicates the end of this column
    flowDataValue    flowDataType -- Choice depends on attribute
}
flowDataType ::= CHOICE {
    flowByteValue    INTEGER (1..255),
    flowShortValue   INTEGER (1..65535),
    flowLongValue    Integer32,
    flowStringValue  OCTET STRING -- Length (n) in first byte,
                        -- n+1 bytes total length, trailing zeroes truncated
}
 ::= { flowColumnActivityEntry 4 }
```

--

-- The Rule Table

--

-- This is an array of rule tables; the one in use is selected by  
 -- CurrentRuleSet. To change the rule set the manager chooses a set  
 -- number which is not in use, downloads the new rule set there, then  
 -- writes the new set number into CurrentRuleSet. Rule set 1 is the  
 -- default rule set, used by the meter on start-up. Several rule sets  
 -- can be held in a meter so that the manager can change the rules  
 -- easily, for example with time of day. Note that a manager may  
 -- not change the default rule set, nor the rules in its current rule  
 -- set! See the 'Traffic Flow Measurement: Architecture' document [9]  
 -- for details of rules and how they are used.

flowRuleTable OBJECT-TYPE

SYNTAX SEQUENCE OF FlowRuleEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Contains all the rule sets which may be used by the meter."

```
::= { flowRules 1 }
```

```

flowRuleEntry OBJECT-TYPE
    SYNTAX FlowRuleEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The rule record itself."
    INDEX { flowRuleSet, flowRuleIndex }
    ::= { flowRuleTable 1 }

FlowRuleEntry ::= SEQUENCE {
    flowRuleSet                INTEGER,
    flowRuleIndex              INTEGER,
    flowRuleSelector           RuleAttributeName,
    flowRuleMask               RuleAddress,
    flowRuleMatchedValue       RuleAddress,
    flowRuleAction              ActionNumber,
    flowRuleParameter          Integer32
}

flowRuleSet OBJECT-TYPE
    SYNTAX INTEGER (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Selects a rule set from the array of rule sets."
    ::= { flowRuleEntry 1 }

flowRuleIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The index into the Rule table.  N.B: These values will
        often be consecutive, given the fall-through semantics of
        processing the table."
    ::= { flowRuleEntry 2 }

flowRuleSelector OBJECT-TYPE
    SYNTAX RuleAttributeName
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Indicates the attribute to be matched.

        null(0) is a special case; null rules always succeed.

        v1(51), v2(52), v3(53), v4(54) and v5(55) select meter
        variables, each of which can hold the name (i.e. selector

```



value) of an address attribute. When one of these is used as a selector, its value specifies the attribute to be tested. Variable values are set by an Assign action."  
 ::= { flowRuleEntry 3 }

flowRuleMask OBJECT-TYPE

SYNTAX RuleAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The initial mask used to compute the desired value. If the mask is zero the rule's test will always succeed."

::= { flowRuleEntry 4 }

flowRuleMatchedValue OBJECT-TYPE

SYNTAX RuleAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The resulting value to be matched for equality. Specifically, if the attribute chosen by the flowRuleSelector logically ANDed with the mask specified by the flowRuleMask equals the value specified in the flowRuleMatchedValue, then continue processing the table entry based on the action specified by the flowRuleAction entry. Otherwise, proceed to the next entry in the rule table."

::= { flowRuleEntry 5 }

flowRuleAction OBJECT-TYPE

SYNTAX ActionNumber

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The action to be taken if this rule's test succeeds, or if the meter's 'test' flag is off. Actions are opcodes for the meter's Packet Matching Engine; details are given in the 'Traffic Flow Measurement: Architecture' document [9]."

::= { flowRuleEntry 6 }

flowRuleParameter OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A parameter value providing extra information for the rule's action."

::= { flowRuleEntry 7 }

```
--
-- Traffic Flow Meter conformance statement
--

flowMIBCompliances
    OBJECT IDENTIFIER ::= { flowMIBConformance 1 }

flowMIBGroups
    OBJECT IDENTIFIER ::= { flowMIBConformance 2 }

flowControlGroup OBJECT-GROUP
    OBJECTS {
        flowRuleInfoSize, flowRuleInfoOwner,
        flowRuleInfoTimeStamp, flowRuleInfoStatus,
        flowInterfaceRate,
        flowInterfaceLostPackets,
        flowReaderTimeout, flowReaderOwner,
        flowReaderLastTime, flowReaderPreviousTime,
        flowReaderStatus,
        flowManagerCurrentRuleSet, flowManagerStandbyRuleSet,
        flowManagerHighWaterMark,
        flowManagerOwner, flowManagerTimeStamp,
        flowManagerStatus,
        flowFloodMark,
        flowInactivityTimeout,
        flowActiveFlows,
        flowMaxFlows }
    STATUS current
    DESCRIPTION
        "The control group defines objects which are used to control
        an accounting meter."
    ::= {flowMIBGroups 1 }

flowDataTableGroup OBJECT-GROUP
    OBJECTS {
        flowDataIndex,
        flowDataStatus,
        flowDataSourceInterface,
        flowDataSourceAdjacentType,
        flowDataSourceAdjacentAddress, flowDataSourceAdjacentMask,
        flowDataSourcePeerType,
        flowDataSourcePeerAddress, flowDataSourcePeerMask,
        flowDataSourceTransType,
        flowDataSourceTransAddress, flowDataSourceTransMask,
        flowDataDestInterface,
        flowDataDestAdjacentType,
        flowDataDestAdjacentAddress, flowDataDestAdjacentMask,
        flowDataDestPeerType,
```

```
flowDataDestPeerAddress, flowDataDestPeerMask,
flowDataDestTransType,
flowDataDestTransAddress, flowDataDestTransMask,
flowDataRuleSet,
flowDataToOctets, flowDataToPDUs,
flowDataFromOctets, flowDataFromPDUs,
flowDataFirstTime, flowDataLastActiveTime,
flowDataSourceClass, flowDataDestClass, flowDataClass,
flowDataSourceKind, flowDataDestKind, flowDataKind
}
```

STATUS current

DESCRIPTION

"The flow table group defines objects which provide the structure for the rule table, including the creation time and activity time indexes into it. In addition it defines objects which provide a base set of flow attributes for the adjacent, peer and transport layers, together with a flow's counters and times. Finally it defines a flow's class and kind attributes, which are set by rule actions."

::= {flowMIBGroups 2 }

flowDataScaleGroup OBJECT-GROUP

```
OBJECTS {
    flowManagerCounterWrap,
    flowDataPDUScale, flowDataOctetScale
}
```

STATUS current

DESCRIPTION

"The flow scale group defines objects which specify scale factors for counters."

::= {flowMIBGroups 3 }

flowDataSubscriberGroup OBJECT-GROUP

```
OBJECTS {
    flowDataSourceSubscriberID, flowDataDestSubscriberID,
    flowDataSessionID
}
```

STATUS current

DESCRIPTION

"The flow subscriber group defines objects which may be used to identify the end point(s) of a flow."

::= {flowMIBGroups 4 }

flowDataColumnTableGroup OBJECT-GROUP

```
OBJECTS {
    flowColumnActivityAttribute,
    flowColumnActivityTime,
    flowColumnActivityIndex,
```

```
        flowColumnActivityData
    }
    STATUS    current
    DESCRIPTION
        "The flow column table group defines objects which can be used
        to collect part of a column of attribute values from the flow
        table."
    ::= {flowMIBGroups 5 }

flowRuleTableGroup OBJECT-GROUP
    OBJECTS {
        flowRuleSelector,
        flowRuleMask, flowRuleMatchedValue,
        flowRuleAction, flowRuleParameter
    }
    STATUS    current
    DESCRIPTION
        "The rule table group defines objects which hold the set(s)
        of rules specifying which traffic flows are to be accounted
        for."
    ::= {flowMIBGroups 6 }

flowMIBCompliance MODULE-COMPLIANCE
    STATUS    current
    DESCRIPTION
        "The compliance statement for a Traffic Flow Meter."
    MODULE
        MANDATORY-GROUPS {
            flowControlGroup,
            flowDataTableGroup,
            flowRuleTableGroup
        }
    ::= { flowMIBCompliances 1 }

END
```

## 5 Acknowledgements

This document was initially produced under the auspices of the IETF's Accounting Working Group with assistance from SNMP and SAAG working groups. Particular thanks are due to Jim Barnes, Sig Handelman and Stephen Stibler for their support and their assistance with checking the MIB.

## 6 References

- [1] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets," STD 17, RFC 1213, Performance Systems International, March 1991.
- [2] Case J., McCloghrie K., Rose M., and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol," RFC 1902, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.
- [3] Case J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for version 2 of the Simple Network Management Protocol SNMPv2", RFC 1903, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.
- [4] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1904, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.
- [5] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework," RFC 1908, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.
- [6] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization, International Standard 8824, December 1987.
- [7] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), International Organization for Standardization, International Standard 8825, December 1987.

[8] Mills, C., Hirsch, G. and G. Ruth, "Internet Accounting Background," RFC 1272, Bolt Beranek and Newman Inc., Meridian Technology Corporation, November 1991.

[9] Brownlee, N., Mills, C., and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2063, The University of Auckland, Bolt Beranek and Newman Inc., GTE Laboratories, Inc, January 1997.

[10] Waldbusser, S., "Remote Network Monitoring Management Information Base, Version 2," Work in Progress.

[11] Reynolds, J., and J. Postel, "Assigned Numbers," STD 2, RFC 1700, ISI, October 1994.

[12] Case, J., "FDDI Management Information Base," RFC 1285, SNMP Research Incorporated, January 1992.

## 7 Security Considerations

Security issues are not discussed in this document.

## 8 Author's Address

Nevil Brownlee  
Information Technology Systems & Services  
The University of Auckland

Phone: +64 9 373 7599 x8941  
EMail: n.brownlee@auckland.ac.nz

